

Platform Service - igloodeveloper Integration

Technical Design Document

Version 1.0

Table of Contents

Platform Service - igloocompany Integration TDD	1
Table of Contents	2
Change Log	3
Objectives	4
Nomenclature	4
Lock Owner	4
Lock User	4
igloohome mobile application (IGH App)	4
Platform Service (PS)	4
Exchange Code	4
Integration Slot	5
Workflows	7
Platform Service - IDP Service Registration	7
Platform Service - IDP Service Token Exchange	8
Access generation - AlgoPIN	9
Event notification via a configured Webhook	10
Example Walkthrough	11

Change Log

Version	Date	Changes
1.0	18 February 2021	First Iteration

Objectives

This document describes the method of integration between a Platform Service provider and igloocompany's igloodeveloper service. The purpose of this integration is to facilitate the Platform Service's users with access generation on igloohome locks within the Platform Service.

Nomenclature:

Lock Owner

The entity that holds the authority over the physical lock. Typically the purchaser of the lock who hands over API access to their lock to the Platform Service.

Lock User

The entity that will be given access to unlock the lock. Typically a user who receives and uses access via the igloohome mobile application.

igloohome mobile application (IGH App)

Android/iOS mobile application that's distributed on the Google Play store and Apple App Store.

Platform Service (PS)

The integration partner's web application (frontend UI) that typically affords their users to make bookings/reservations for some asset/resource.

Exchange Code

Short-lived and unique alphanumeric code that is generated and shown on the *IGH App* for a specific lock. The *Exchange Code* is to be used on the *PS* to exchange for an API token. Once used, the *Lock Owner* will no longer be able to view the *Exchange Code* on the *IGH App*.

Integration Slot

A prerequisite virtual object, tied to an igloohome account, that is necessary for integration with Platform Service's software services. At least one *Integration Slot* is required to integrate with the *PS*. Should the *Lock Owner* desire to simultaneously integrate multiple igloohome locks with the *PS*, multiple *Integration Slots* are required.

An *Integration Slot* is considered "occupied" when the lock's *Exchange Code* has been used to associate the *Integration Slot* with the lock. Only one lock can occupy an *Integration Slot* at any point in time.

Deleting one/all *Integration Slot(s)*

To perform this action, the *Lock Owner* will have to:

1. Navigate to the integration section on the *IGH App*
2. Unsubscribe to one/all *Integration Slots*

Note: either one of these actions will invalidate the corresponding API token! *PS* should observe these webhook events to react appropriately.

Removing of locks occupying the *Integration Slot*

To perform this action, the *Lock Owner* could have:

- Vacated the lock from the *Integration Slot* via the *IGH App*
- Or
- Unpaired the lock using the *IGH App*

Note: either one of these actions will invalidate the corresponding API token! *PS* should observe these webhook events to react appropriately.

Swapping of locks occupying the *Integration Slot*

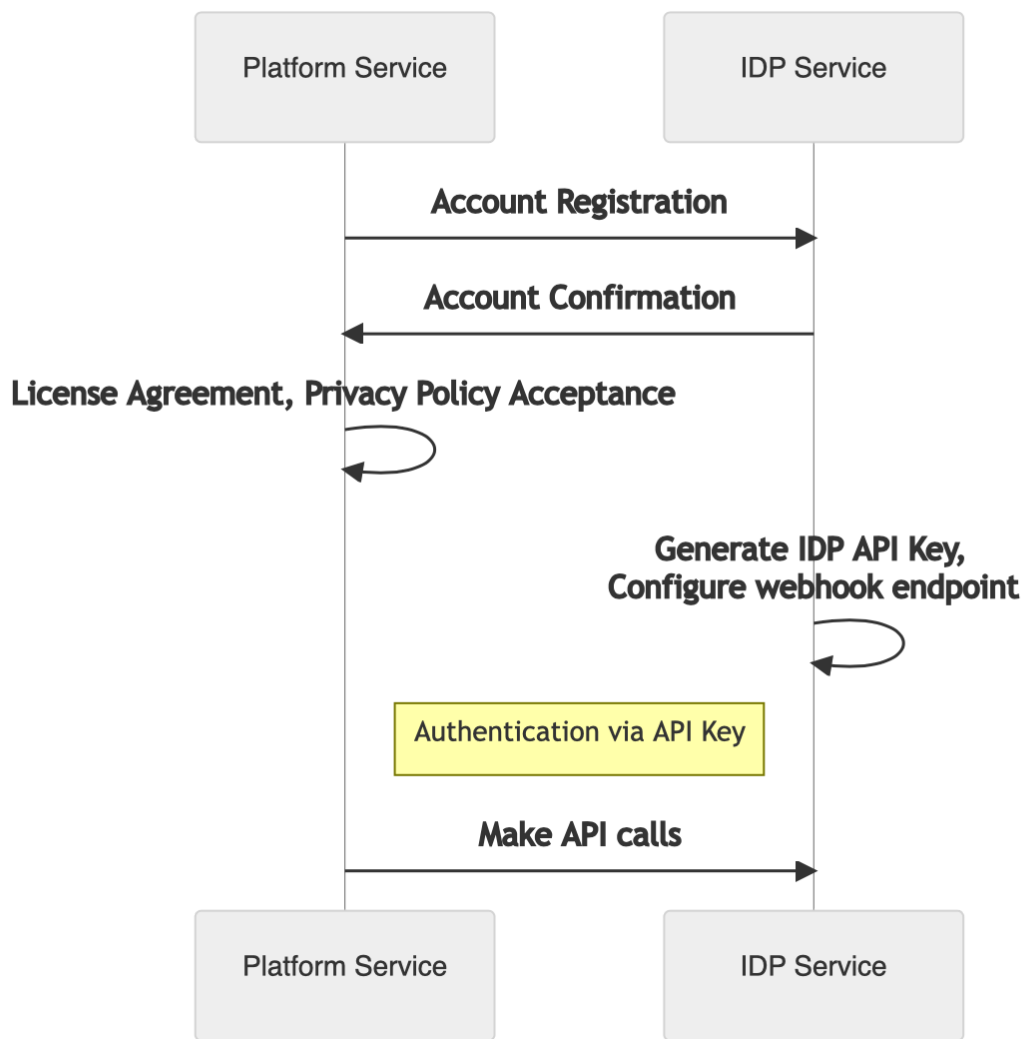
To perform this action, the *Lock Owner* will have to:

1. Vacate the lock from the *Integration Slot* via the *IGH App*
2. Navigate to the new lock s/he wishes to occupy the new *Integration Slot*

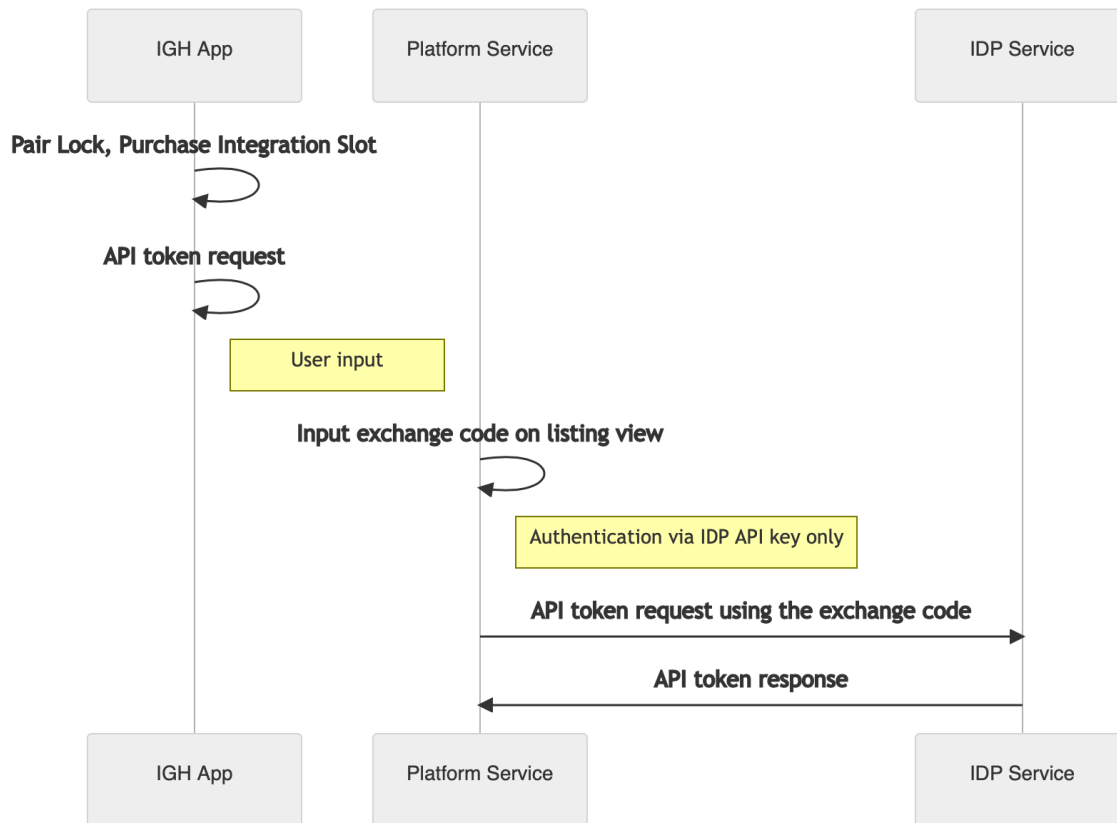
3. Generate an *Exchange Code* for the lock
4. Use the *Exchange Code* on the *PS*

Workflows

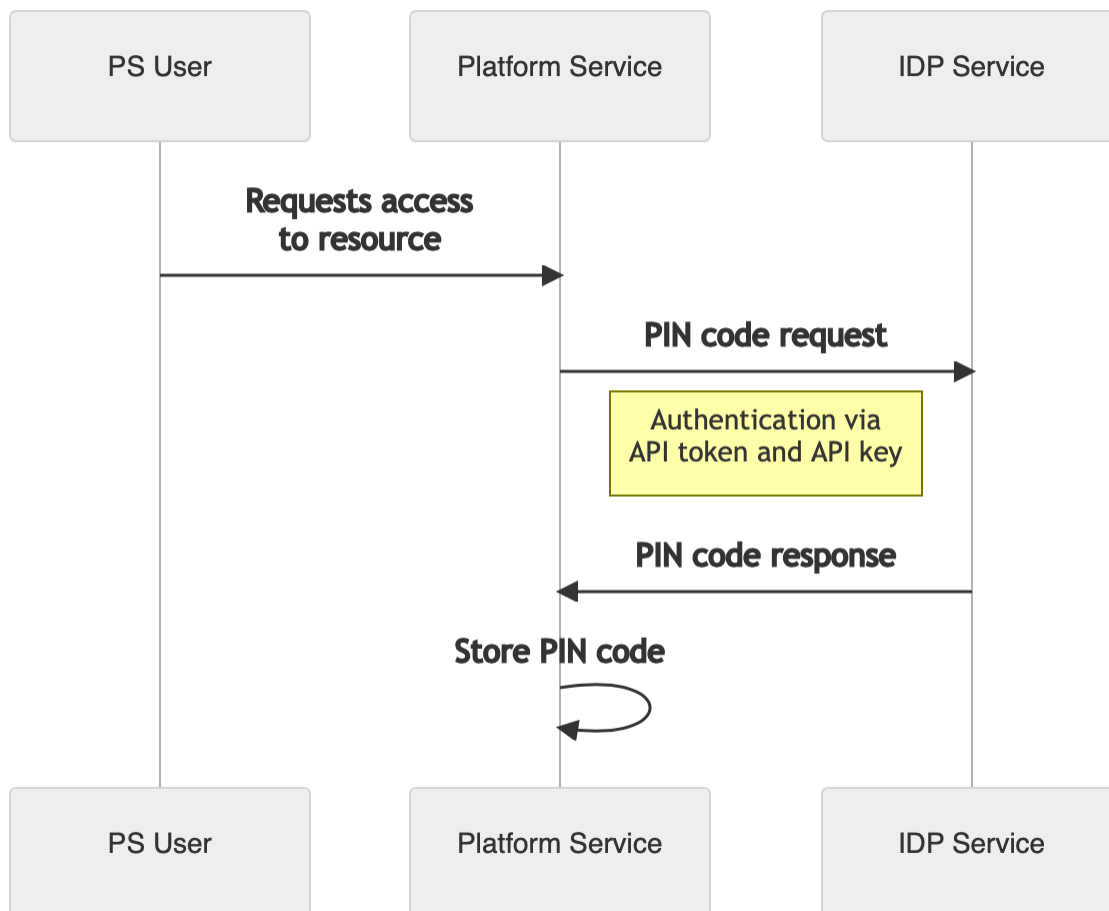
Platform Service - IDP Service Registration (One time)



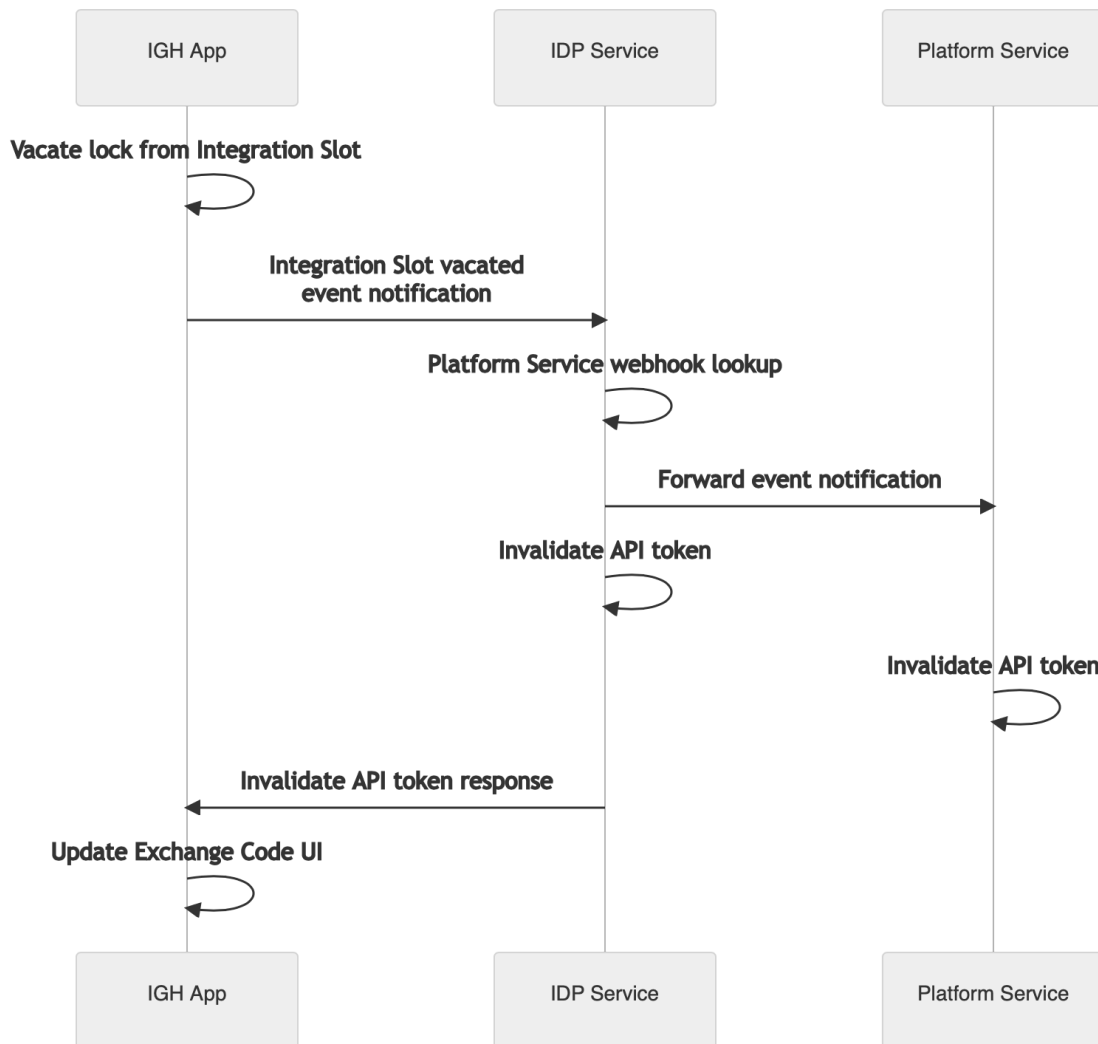
Platform Service - IDP Service Token Exchange



Access generation - AlgoPIN



Event notification via a configured Webhook



Example Walkthrough

The overall process can be divided into the following steps:

1. Platform Service registers an account on the igloodeveloper portal
2. Platform Service generates an igloodeveloper API key
3. Platform Service configures the event notification webhook endpoint on the igloodeveloper portal
4. The *Lock Owner* downloads the *IGH App*
5. The *Lock Owner* creates an igloohome account on the *IGH App*
6. The *Lock Owner* pairs/provisions an igloohome lock using the *IGH App*
7. The *Lock Owner* navigates to the integration section on the *IGH App*, enters their billing information and makes a purchase for an *Integration Slot*
8. The *Lock Owner* navigates to the lock of choice for integration on the *IGH App* and generates the short-lived *Exchange Code* to be used on the *PS*.
9. The *Lock Owner* logs in to the *PS*, creates/edits a resource, and enters the *Exchange Code* (shown on the *IGH App*) into a text field.
10. *PS* makes a HTTP request to the igloodeveloper API token endpoint using the API key (from Step 2) as authentication along with the *Exchange Code* (from Step 8).
11. igloodeveloper service will send a HTTP response with an API token to be stored by the *PS* to use in conjunction with the API key as authentication for all future HTTP requests. At this point, the integration is complete and the *Exchange Code* will no longer be available/shown on the *IGH App*.
12. When a *Lock User* makes a booking/reservation, the *PS* will make a HTTP request to the igloodeveloper AlgoPIN generation endpoint. This request is authenticated via the igloodeveloper API token and API key generated earlier. The HTTP response containing the AlgoPIN should be retained on the *PS*' database for distribution.
13. The *Lock User* can now use the AlgoPIN code to unlock the lock.
14. Fast-forward to some point in the future, and assume the *Lock Owner* has either decided to delete one of their *Integration Slots* or remove a lock from an *Integration Slot* that is currently occupied. The corresponding (delete or remove)

igloocompany

webhook event will be forwarded to the webhook endpoint configured in Step 3. The API token for the lock is no longer valid and should be cleaned up on both the igloodeveloper service as well as the *PS*.